IN THE CLAIMS:

This listing of claims will replace all prior versions, and listing of claims, in the Application.

Listing of claims:

1. (Currently amended) A software system for constrained graphs~~, the system implemented in accordance with an object oriented design framework, wherein said graph is constructed using a plurality of graphical elements, the system~~ comprising:

software code for implementing a graph, said graph being constructed using a plurality of subgraphs having each a pre-defined grouping of a plurality of graphical elements;

software code for repositioning elements of a subgraph; and

software code for repositioning other subgraphs when the other subgraphs are affected by the repositioning of the elements of the subgraph.

~~a plurality of subgraph classes, wherein an instance of each of said subgraph classes comprises a predefined grouping of one or more of said graphical elements representative of a subgraph type; and~~

~~each of said plurality of subgraph classes adapted to:~~

~~reposition the graphical elements of a subgraph within said graph, said subgraph represented by an instance of one of said plurality of subgraph classes; and~~

~~initiate a repositioning of the graphical elements of subgraphs affected by said repositioning of the graphical elements of the subgraph represented by said instance of said one of said plurality of subgraph classes.~~

2. (Currently amended) The software system of claim 1[[,]] ~~wherein each of said plurality of subgraph classes is~~ further ~~adapted to:~~ comprising software code for displaying the graphical elements of each subgraph in a specified layout format.

~~display the graphical elements of a subgraph represented by said instance of said one of said plurality of subgraph classes to a user in a specified layout format.~~

3. (Original) The software system of claim 2, wherein said specified layout format comprises a layout selected from the following group: a horizontal layout and a vertical layout.

4. (Original) The software system of claim 2, wherein said specified layout comprises a directional layout.

5. Canceled.

6. (Currently amended) The software system of claim 4 ~~0~~, wherein the affected subgraphs are repositioned by repositioning their graphical elements ~~said first subgraph class comprises an abstract class~~.

7. (Currently amended) The software system of claim 1, further comprising:

a layout manager for ~~adapted to~~:

~~initiate~~ initiating the repositioning and display of the graphical elements of a plurality of subgraphs in said graph by commanding the repositioning and display of the graphical elements of a selected subgraph in said graph.

8. (Currently amended) The software system of claim 7 wherein said layout manager ~~is~~ further ~~adapted to~~:

~~identify~~ identifies a plurality of subgraphs in said graph;

~~receive~~ receives an identifier of an input subgraph in said graph;

~~determine~~ determines from said identifier a selected subgraph to be shifted; and

~~command~~ commands said selected subgraph to reposition and display the graphical elements.

9. (Currently amended) The software system of claim 8 further comprising:

a first layout manager class which ~~can be~~ when extended ~~to define~~ defines one or more second layout manager classes; and

wherein said first layout manager comprises an instance of a second layout manager class.

10. (Original) The software system of claim 8, wherein the selected subgraph determined from said identifier comprises said input subgraph.

11. (Original) The software system of claim 9, wherein said instance of a second layout manager class is created when one or more graphical elements are added to or deleted from said graph.

12. (Original) The software system of claim 11, further comprising a module for obtaining input from a user, wherein a request to add or delete graphical elements from said graph is generated from said input.

13. (Original) The software system of claim 8, wherein data associated with subgraphs identified by an instance of said second layout manager class is stored in a map, and wherein said map is used by instances of said second subgraph classes in determining affected subgraphs.

14. (Original) The software system of claim 13, wherein said map comprises a hash map.

15. (Currently amended) The software system of claim 1, wherein said repositioning of the graphical elements of said ~~specific~~ subgraph requires that said graphical elements be shifted either horizontally or vertically in said graph.

16. (Original) The software system of claim 2, wherein said specified layout comprises a directional layout.

17. (Original) The software system of claim 1, wherein a subgraph comprises a further subgraph.

18. Canceled.

19. (Currently amended) A software system for use in the design of software applications in which a constrained graph is displayed, the system implemented in accordance with an object-oriented design framework, wherein said graph is constructed using a plurality of graphical elements, the system comprising:

a first subgraph class, wherein said first subgraph class ~~can be~~ is extended to define a plurality of second subgraph classes, wherein an instance of each of said second subgraph classes represents a subgraph of a specific subgraph type, wherein each subgraph of a specific subgraph type is composed of a predefined grouping of ~~one or more of~~ said graphical elements, and wherein each of said plurality of second subgraph classes implements one or more first methods for:

repositioning the graphical elements of a subgraph represented by an instance thereof within said graph and determining affected subgraphs,

displaying the graphical elements of a subgraph represented by an instance thereof to ~~said~~ a user in a specified layout format, and

commanding a repositioning and display of the ~~graphical elements of said~~ affected subgraphs.

20. (Currently amended) The software system of claim 19, further comprising a first layout manager class, wherein said first layout manager class ~~can be~~ is extended to define one or more second layout manager classes, wherein an instance of each of said second layout manager classes

represents a layout manager, wherein each of said second layout manager classes implements one or more second methods for:

identifying a plurality of subgraphs in said graph,

receiving an identifier of an input subgraph in said graph,

determining from said identifier a selected subgraph to be shifted, and

commanding a repositioning and display of the graphical elements of said selected subgraph by calling the one or more first methods implemented by the second subgraph class of which said selected subgraph is an instance;

such that when an instance of a second layout manager class is created, said one or more second methods are executed, whereby the layout manager represented by said instance identifies a plurality of subgraphs in said graph and initiates the repositioning and display of the graphical elements of a plurality of subgraphs in said graph by commanding the repositioning and display of the graphical elements of a selected subgraph in said graph.

21. (Currently amended) The system as claimed in claim 19, wherein the affected subgraphs are repositioned by repositioning their graphical elements ~~said first subgraph class is an abstract class~~.

22. (Original) The system as claimed in claim 20, wherein the selected subgraph determined from said identifier is said input subgraph.

23. (Original) The system as claimed in claim 19, wherein said instance of a second layout manager class is created when one or more graphical elements are added to or deleted from said graph.

24. (Original) The system as claimed in claim 23, further comprising a module for obtaining input from a user, wherein a request to add or delete graphical elements from said graph is generated from said input.

25. (Original) The system as claimed in claim 19, wherein data associated with subgraphs identified by an instance of said second layout manager class is stored in a map, and wherein said map is used by instances of said second subgraph classes in determining affected subgraphs.

26. (Currently amended) A computer readable media storing data and instructions, said data and instructions when executed by a computing device ~~adapt~~ cause said computing device to:

implement a graph, the graph including a plurality of subgraphs having each a predefined grouping of a plurality of graphical elements;

~~organize a plurality of subgraph classes, wherein an instance of each of said subgraph classes comprises a predefined grouping of one or more of said graphical elements representative of a subgraph type; and~~

~~each of said plurality of subgraph classes adapted to:~~

reposition the graphical elements of a subgraph within said graph~~, said subgraph represented by an instance of one of said plurality of subgraph classes~~; and

initiate a repositioning of ~~the graphical elements of~~ subgraphs affected by said repositioning of the graphical elements of the subgraph ~~represented by said instance of said one of said plurality of subgraph classes~~.

27. (Currently amended) The computer readable media of claim 26, wherein each of said plurality of subgraph ~~classes~~ subgraphs is ~~further adapted to:~~

~~display~~ displayed ~~the graphical elements of a subgraph represented by said instance of said one of said plurality of subgraph classes to a user~~ in a specified layout format.

28. (Original) The computer readable media of claim 26, wherein said specified layout format comprises a layout selected from the following group: a horizontal layout and a vertical layout.

29. (Original) The computer readable media of claim 26, wherein said specified layout comprises a directional layout.

30. Canceled.

31. (Currently amended) The computer readable media of claim 26 ~~30~~, wherein the affected subgraphs are repositioned by repositioning their graphical elements ~~said first subgraph class comprises an abstract class~~.

32. (Currently amended) The computer readable media of claim 0, wherein the executed data and instructions further ~~adapting~~ cause said computer device to:

organize a layout manager, the layout manager ~~adapted to:~~

~~initiate~~ initiating the repositioning and display of the graphical elements of a plurality of subgraphs in said graph by commanding the repositioning and display of the graphical elements of a selected subgraph in said graph.

33. (Currently amended) The computer readable media of claim 32 wherein said layout manager ~~is~~ further ~~adapted to~~:

~~identify~~ identifies a plurality of subgraphs in said graph;

~~receive~~ receives an identifier of an input subgraph in said graph;

~~determine~~ determines from said identifier a selected subgraph to be shifted; and

~~command~~ commands said selected subgraph to reposition and display the graphical elements.

34. Canceled.

35. (Original) The computer readable media of claim 33, wherein the selected subgraph determined from said identifier comprises said input subgraph.

36. Canceled.

37. (Currently amended) The computer readable media of claim 36, wherein the executed data and instructions further cause ~~further adapting~~ said computer device to organize a module for obtaining input from a user,

wherein a request to add or delete graphical elements from said graph is generated from said input.

38. Canceled.

39. Canceled.

40. (Original) The computer readable media of claim 26, wherein said repositioning of the graphical elements of said specific subgraph requires that said graphical elements be shifted either horizontally or vertically in said graph.

41. (Original) The computer readable media of claim 26, wherein said specified layout comprises a directional layout.

42. (Original) The computer readable media claim 26, wherein a subgraph comprises a further subgraph.

43. (Currently amended) A layout manager defined by a layout manager interface, said layout manager interface provided by a software system for use in the design of software applications in which a constrained graph is displayed to a user, ~~the system implemented in accordance with an object oriented design framework, wherein said graph is constructed using a plurality of graphical elements,~~ the layout manager ~~system~~ comprising:

~~a first subgraph class, wherein said first subgraph class can be extended to define a plurality of second subgraph classes, wherein an instance of each of said second subgraph classes represents a subgraph of a specific subgraph type, wherein each subgraph of a specific subgraph type is composed of a predefined grouping of one or more of said graphical~~

~~elements, and wherein each of said plurality of second subgraph classes implements one or more first methods for~~

~~repositioning the graphical elements of a subgraph represented by an instance thereof within said graph and determining affected subgraphs.~~

~~displaying the graphical elements of a subgraph represented by an instance thereof to said user in a specified layout format, and~~

~~commanding a repositioning and display of the graphical elements of said affected subgraphs; and~~

a first layout manager class ~~interface~~, wherein said first layout manager class is ~~can be~~ extended to define one or more second layout manager classes, wherein an instance of each of said one or more second layout manager classes represents a layout manager, wherein each of said one or more second layout manager classes implements a method ~~one or more second methods for~~ comprising:

identifying a plurality of subgraphs in said graph,

receiving an identifier of an input subgraph in said graph,

determining from said identifier a selected subgraph to be shifted, ~~and~~

commanding a repositioning and display of the graphical elements of said selected subgraph by calling the method ~~one or more first methods~~ implemented by the second

subgraph class of which said selected subgraph is an instance[[;]].

determining other subgraphs affected by the repositioning of the graphical elements of said selected subgraph, and

commanding a repositioning and display of the affected subgraphs.

~~such that when an instance of a second layout manager class is created, said one or more second methods are executed, whereby layout manager represented by that instance identifies a plurality of subgraphs in said graph and initiates the repositioning and display of the graphical elements of a plurality of subgraphs in said graph by commanding the repositioning and display of the graphical elements of a selected subgraph in said graph.~~

44. (Currently amended) A method of displaying a constrained graph, said constrained graph being constructed by a plurality of subgraphs having each a predefined grouping of a plurality of graphical elements ~~using a plurality of graphical elements, wherein a first subgraph class is defined that can be extended to define a plurality of second subgraph classes, wherein an instance of each of said second subgraph classes represents a subgraph of a specific subgraph type, wherein each subgraph of a specific subgraph type is composed of a predefined grouping of one or more of said graphical elements, and wherein each of said plurality of second subgraph classes implements one or more first methods for repositioning the graphical elements of a subgraph represented by an instance thereof within said graph and determining affected subgraphs,~~

~~displaying the graphical elements of a subgraph represented by an instance thereof to said user in a specified layout format, and commanding a repositioning and display of the graphical elements of said affected subgraphs~~, said method comprising:

~~identifying a plurality of subgraphs in said graph;~~

receiving from a user ~~an identifier of~~ an input for deleting at least one graphical element from or adding at least one graphical element to a particular subgraph ~~in said graph~~;

determining whether to reposition one or more graphical elements from the predefined grouping of the graphical elements of the particular subgraph in response to the addition or deletion of the at least one graphical element ~~from said identifier a selected subgraph to be shifted~~; ~~and~~

repositioning the one or more graphical elements of the particular subgraph if it is determined that the one or more graphical elements from the predefined grouping of the graphical elements are to be repositioned;

determining whether location of one or more subgraphs is affected by the repositioning of the one or more graphical elements of the particular subgraph; and

repositioning, if one or more subgraphs are affected, the one or more affected subgraphs.

~~commanding a repositioning and display of the graphical elements of said selected subgraph by calling the one or more first methods implemented~~

~~by the second subgraph class of which said selected subgraph is an instance;~~

~~whereby a plurality of subgraphs in said graph are identified, and the repositioning and display of the graphical elements of a plurality of subgraphs in said graph is initiated by commanding the repositioning and display of the graphical elements of a selected subgraph in said graph.~~

45. Canceled.

46. Canceled.

47. Canceled.

48. Canceled.

49. (Currently amended) A method of displaying a constrained graph, said graph comprising a plurality of graphical elements and a plurality of subgraphs, wherein each of said plurality of subgraphs comprises a grouping ~~of one or more~~ of said graphical elements, said method comprising:

determining from an identifier of an input subgraph in said graph, a selected subgraph to be repositioned; and

repositioning the ~~graphical elements of said~~ selected subgraph.

50. (Currently amended) A method of displaying a constrained graph, said graph comprising a plurality of graphical elements and a plurality of subgraphs, wherein each of said plurality of subgraphs comprises a

grouping of ~~one or more of~~ said graphical elements, said method comprising:

repositioning the graphical elements of a subgraph within said graph; and

initiate a repositioning of ~~the graphical elements of~~ subgraphs affected by said repositioning of the graphical elements of said subgraph.

51. (Currently amended) The method of claim 50, wherein each of said plurality of subgraphs ~~is adapted to display~~ displays the graphical elements of a subgraph in a specified layout format.

52. (Original) The method of claim 50, further comprising:

initiating the repositioning and display of the graphical elements of a plurality of subgraphs in said graph by commanding the repositioning and display of the graphical elements of a selected subgraph in said graph.

53. (Original) The method of claim 0 further comprising:

identifying a plurality of subgraphs in said graph;

receiving an identifier of an input subgraph in said graph;

determining from said identifier a selected subgraph to be shifted; and

commanding said selected subgraph to reposition and display the graphical elements.